



学生应知信息知识

Delphi 简明教程 (下)

秋登峰 主编

目 录

第七章	管理文件、目录和驱动器	1
7.1	各种控件简介	1
7.1.1	驱动器控件	2
7.1.2	目录控件	5
7.1.3	文件列表控件	7
7.1.4	文件过滤器控件	10
7.2	利用控件管理文件	14
7.3	管理目录	22
7.3.1	ChDir 函数	23
7.3.2	Mkdir 函数	23
7.3.3	Rmdir 函数	24
7.3.4	应用实例	25
7.4	管理驱动器	33
7.4.1	DiskSize 函数	33
7.4.2	DiskFree 函数	36
7.4.3	GetDir 函数	37
7.5	小 结	38
第八章	数据库开发和管理	38
8.1	使用数据库向导	39
8.2	自己定制数据库	48
8.3	数据库管理程序	63
8.4	设计数据库报表	74
8.5	小 结	78
第九章	多媒体的设计和使用	78
9.1	认识 MediaPlayer 控件	79
9.1.1	AutoEnable 属性	81
9.1.2	Display 属性	84

9.1.3	Filename 属性.....	87
9.1.4	DeviceType 属性.....	88
9.1.5	EnabledButtons 属性.....	89
9.2	音频播放程序.....	91
9.3	视频播放程序.....	105
9.4	小 结.....	122

第七章 管理文件、目录和驱动器

所谓的格式文件是指存放在外部存储介质上的数据和程序等，在 Windows 的操作环境中，文件是一个非常重要的概念。

为了方便用户存取和操作的方便，在 Delphi 中带有强大的文件管理功能，在其中用户可以通过调用文件处理函数来对文件和目录进行各种各样的处理和操作，如创建目录，打开和删除文件等操作。

7.1 各种控件简介

在本章中，将讨论以下几个问题：

- 如何使用常用的文件类 ActiveX 控件
- 如何利用文件类控件来设计文件系统的界面
- 利用文件类的函数来进行文件内部的操作

下面首先来介绍一下文件类 ActiveX 控件的使用方法。

Delphi 中的文件类控件包括 DirectoryListBox 控件（目录列表框控件）、FileListBox 控件（文件列表框控件）、DriveComboBox 控件（驱动器组合列表框控件）和 FilterComboBox 控件（过滤组合列表框控件）四种。

如图 7-1 所示即为一个装有以上四种文件类控件的窗体。

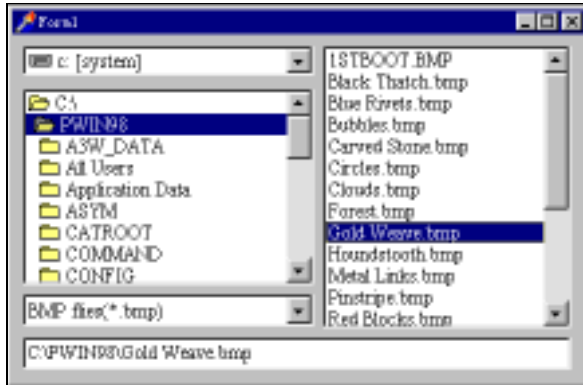


图 7-1 放置文件类控件后的窗体

在图 7-1 中，这四个文件类控件之间的关系必须是协调的，如 DriveComboBox 控件中显示的是 C 驱动器，则 DirectoryListBox 控件也必须显示的是“C:\Pwin98”，FileListBox 控件中显示的是“C:\Pwin98”中的 BMP 文件内容。实际上，实现这些协调，需要用户编制程序才能实现，下面我们就来一起学习。

7.1.1 驱动器控件

DriveComboBox 控件（驱动器组合列表框控件）的功能是显示当前计算机中全部有效的驱动器，以供用户选择和操作。

DriveComboBox 控件的属性在这里就不一一的加以介绍了，下面以一个示例来说明 DriveComboBox 控件和它的属性的用法，具体步骤如下所示：

1. 开始工作

首先启动一个新的项目，选择菜单“File”中的“New Application”项，在 Delphi 的集成开发环境中就会弹出一个新建的窗体，在控件工具栏上选择“Win32”选项

后，在“ DriveComboBox 控件 ”的图标上双击鼠标的左键，这时空白的窗体上就会出现一个 DriveComboBox 控件，如图 7-2 所示。

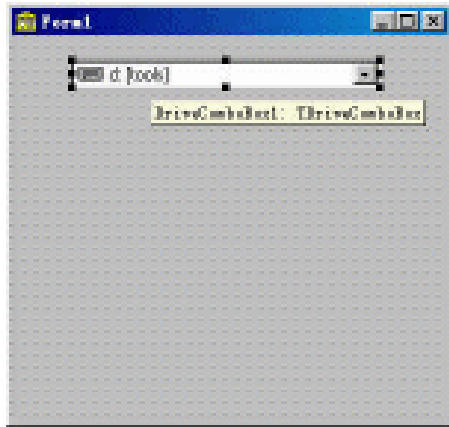


图 7-2 添加控件后的窗体

2. 添加初始化代码

添加控件后，用鼠标的左键在窗体上的空白处双击，把光标移动到代码窗口中的适当位置，找到需要添加代码的事件——窗体的 FormCreate()事件，在其中添加如下代码：

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
    drivecombobox1.Drive:='d';  
    //设置驱动器为 D 盘  
    drivecombobox1.TextCase:=tcUpperCase;  
    //设置显示的字母为大写字母  
end;
```

程序说明：

DriveComboBox 控件的 Drive 用于设置控件显示的驱动器的盘符，所以语句 drivecombobox1.Drive:='d';的

作用是设置控件显示的驱动器为 D 盘；

而 TextCase 属性用于设置 DriveComboBox 控件中的字符显示形式，所以语句 `drivecombobox1.TextCase:=tcUpperCase;` 的作用是设置控件显示的字符为大写字符。

在以上的程序段中，我们通过语句 `drivecombobox1.TextCase:=tcUpperCase;` 来设置 DriveComboBox 控件中显示的字符为大写字符，那么可不可以通过程序来设置 DriveComboBox 控件中显示为小写字符呢？

提示：

与设置显示字符为大写字符一样，设置 DriveComboBox 控件中显示为小写字符的语句为 `drivecombobox1.TextCase:=tcLowerCase;`。

3. 运行程序

做完以上的工作后，存储文件，按键盘上的功能键 F9 运行程序，程序的运行结果如图 7-3 所示。

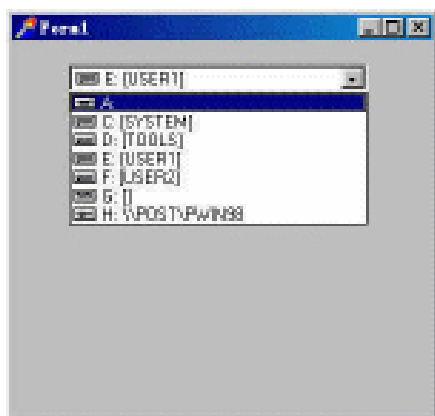


图 7-3 程序运行结果

7.1.2 目录控件

DirectoryListBox 控件(目录列表框控件)的功能是显示当前计算机有效驱动器中当前驱动器的树形目录结构,以供用户选择和操作,用户在选择的过程中可以在同一个驱动器的不同目录之间进行切换等操作。

DirectoryListBox 控件的属性在这里就不一一的加以介绍了,如果读者有什么不清楚的地方,请参考 DriveComboBox 控件即可。

下面以一个示例来说明 DirectoryListBox 控件控件和它的属性的用法,具体步骤如下所示:

1. 开始工作

首先启动一个新的项目,选择菜单“File”中的“New Application”项,在 Delphi 的集成开发环境中就会弹出一个新建的窗体,在控件工具栏上选择“Win3.1”选项后,在“DirectoryListBox 控件”的图标上双击鼠标的左键,这时空白的窗体上就会出现一个 DirectoryListBox 控件,然后在控件工具栏上选择“Standard”选项,在“Label 控件”的图标上双击鼠标的左键,在窗体上就会添加一个 Label 控件。

添加控件后的窗体如图 7-4 所示。

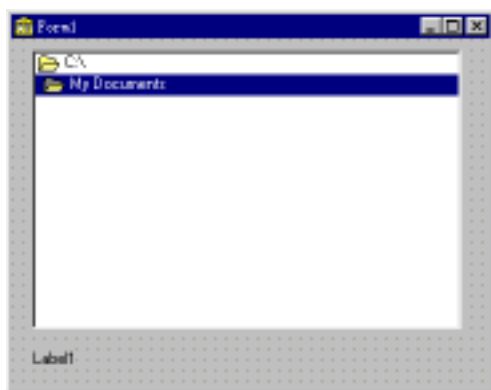


图 7-4 添加控件后的窗体

2. 添加代码

当然，控件和窗体的属性设置不但可以在程序的设计过程中由用户手动设置，还可以在程序的运行过程中有程序控制，如以上的两个控件的属性设置可以通过以下的程序段来实现。

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  chdir('c:\mydocument');
  //设置系统的缺省路径为 c:\mydocument
  directorylistbox1.Align:=alTop;
  //DirectoryListBox 控件始终位于窗体的顶部
  directorylistbox1.DirLabel:=label1;
  //与 DirectoryListBox1 相关联的控件为 label1
end;
end.
```

3. 运行程序

做完以上的工作后，按键盘上的功能键 F9 运行程序，程序的运行结果如图 7-5 所示。

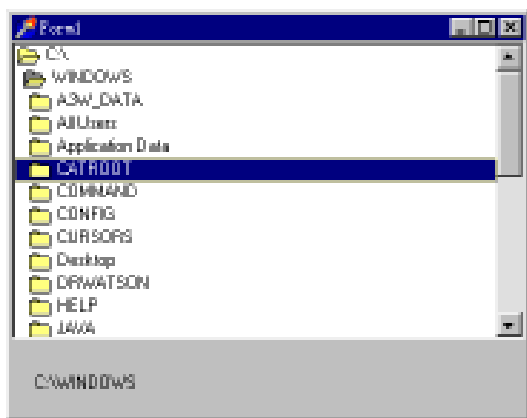


图 7-5 程序运行结果

在程序的运行过程中，可以选择不同的目录，同时在 Label 控件中会显示出用户在程序运行过程中所选择的路径。

7.1.3 文件列表控件

FileListBox 控件（文件列表框控件）的功能是显示当前路径中符合条件的所有文件列表，以供用户选择和操作，用户在选择的过程中可以在同一个路径中的不同文件之间进行切换等操作。

FileListBox 控件的属性在这里就不一一的加以介绍了，如果读者有什么不清楚的地方，请参考有关的参考手册，下面以一个示例来说明 FileListBox 控件和它的属性的用法，另外还介绍了与 FileListBox 控件相关联的 Edit 控件，其中 Edit 控件用于显示 FileListBox 控件中的文件名。

制作示例程序的具体步骤如下所示：

1. 开始工作

首先启动一个新的项目，选择菜单“File”中的“New

Application”项，在 Delphi 的集成开发环境中就会弹出一个新建的窗体，在控件工具箱上选择 FileListBox 控件的图标、Edit 控件的图标和 Button 控件的图标，并且把它们拖放到空白的窗体上，添加控件后的窗体如图 7-6 所示。

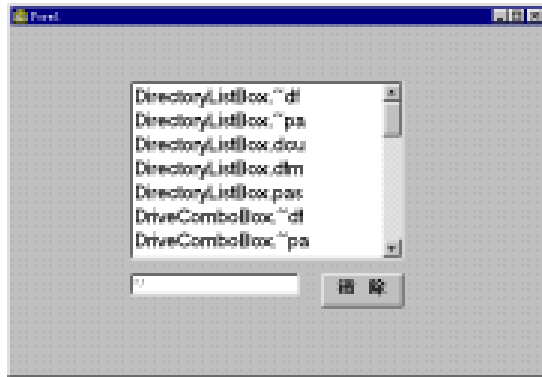


图 7-6 添加控件后的窗体

2. 添加程序代码

在本程序中需要添加代码的事件有窗体的 FormCreate() 事件，按钮 Button1MouseDown() 事件、Button1MouseUp() 事件和 FileListBox 控件的 FileListBox1Change() 事件，在这里只以窗体的 FormCreate() 事件为例来说明添加代码的过程，其它事件的响应代码请读者参见附后的源程序。

在窗体的事件列表框中选择 FormCreate() 事件，这时在代码窗口中就会显示出 FormCreate() 事件的框架，将光标定位到代码窗口中，并且移动到需要添加代码的位置，然后添加下列代码：

```
procedure TForm1.FormCreate(Sender: TObject);  
  
begin  
  
filelistbox1.Drive:='d';
```

```
//设置驱动器  
filelistbox1.Directory:='d:\delphi\bin';  
  
//设置文件显示路径  
filelistbox1.FileEdit:=edit1;  
  
//设置关联控件  
end;
```

窗体的 `FormCreate()` 事件在窗体的装入的同时就会被激活，进入到窗体 `FormCreate()` 事件的过程中，首先通过语句 `filelistbox1.Drive:='d'`；设置系统默认的驱动器为 D 盘，然后设置文件的显示路径为 `d:\delphi\bin`，最后通过语句 `filelistbox1.FileEdit:=edit1`；来设置 `Edit1` 控件为 `FileListBox1` 控件的关联控件。

3. 运行程序

做完以上的工作后，按键盘上的功能键 `F9` 运行程序，程序的运行结果如图 7-7 所示。

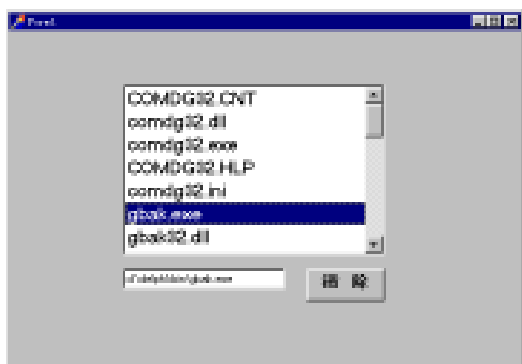


图 7-7 程序运行结果

在程序的运行过程中，当用户在 `FileListBox1` 控件中选择一个有效的文件时，在 `Edit1` 控件中就会显示出选中文件的路径和文件名，如果用户在“清除”按钮上按下鼠标的左键时，在 `FileListBox1` 控件中就会清除文

件列表，但是只要抬起鼠标，在 FileListBox1 控件中又会重新显示指定路径下的文件列表，如图 7-8 所示。



图 7-8 清除文件列表框中的内容

7.1.4 文件过滤器控件

Delphi 中常用的文件类控件中最后一种控件为 FilterComboBox 控件，作用是为用户提供从文件列表框中筛选文件的条件，它的作用决定了 FilterComboBox 控件要同其它几种文件类控件相互协调才能够完成文件的筛选功能。

FilterComboBox 控件的属性在这里就不一一的加以介绍了，如果读者有什么不清楚的地方，请参考有关的参考手册，下面以一个示例来说明 FilterComboBox 控件和它的属性的用法，另外还介绍了如何使用 FilterComboBox 控件与 FileListBox 控件相互协调工作。制作示例程序的具体步骤如下所示：

1. 开始工作

首先启动一个新的项目，选择菜单“File”中的“New Application”项；

然后,在 Delphi 的集成开发环境中就会弹出一个新建的窗体,选择 FileListBox 控件和 FilterComboBox 控件,并且把它们拖放到空白的窗体上,添加控件后的窗体如图 7-9 所示。

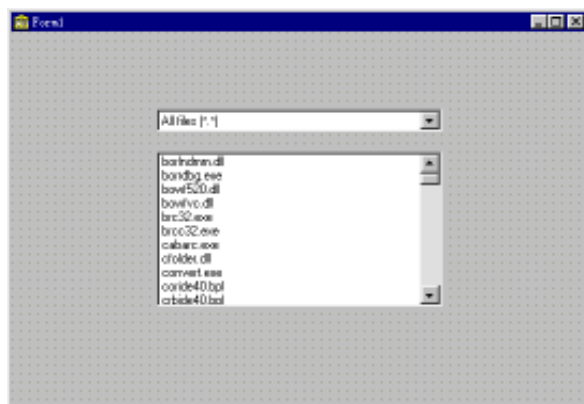



图 7-9 添加控件后的窗体

2. 过滤器设置

在程序的设计阶段,用鼠标的左键单击 FilterComboBox 控件属性列表中的 Filter 属性设置框中的按钮 ,就会弹出一个如图 7-10 所示的过滤器设置对话框,用户可以在其中设置文件显示的过滤器。

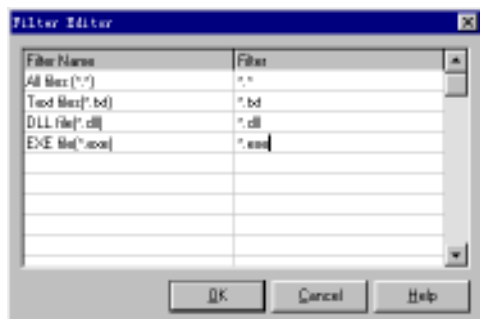


图 7-10 过滤器设置

在本程序中,过滤器的属性设置为 All files

(*.*)|*.*|Text files(*.txt)|*.txt|DLL file(*.dll)|*.dll|EXE file(*.exe)|*.exe，也就是说，在程序的运行过程中，用户在显示文件类型上有四种选择：

- 显示全部文件；
- 显示以*.txt 结尾的文本文件；
- 显示以*.dll 结尾的动态链接库文件；
- 显示以*.exe 结尾的可执行文件。

3. 响应窗体事件

在程序的设计阶段，用鼠标的左键双击窗体的空白处，在屏幕上就会弹出一个代码窗口，在缺省的情况下，代码窗口中显示的是窗体的 FormCreate()事件，将光标移动代码窗口中，并且添加下列代码：

```
procedure TForm1.FormCreate(Sender: TObject);  
  
begin  
  
filtercombobox1.FileList:=filelistbox1;  
  
//设置关联控件  
  
filelistbox1.Drive:='d';  
  
//设置控件的驱动器  
  
filelistbox1.Directory:='d:\delphi\bin'  
  
//设置控件的文件路径  
  
end;  
  
end.
```

窗体的 FormCreate()事件在窗体的装入的同时就会被激活，进入到窗体的 FormCreate()事件的过程中，首先通过语句 filtercombobox1.FileList:=filelistbox1;FileListBox1 控件为 FilterComboBox 控件的关联控件，然后设置文件列表框控件的文件路径为 d:\delphi\bin。

4. 运行程序

做完以上的工作后，按键盘上的功能键 F9 运行程序，程序的运行结果如图 7-11 所示。

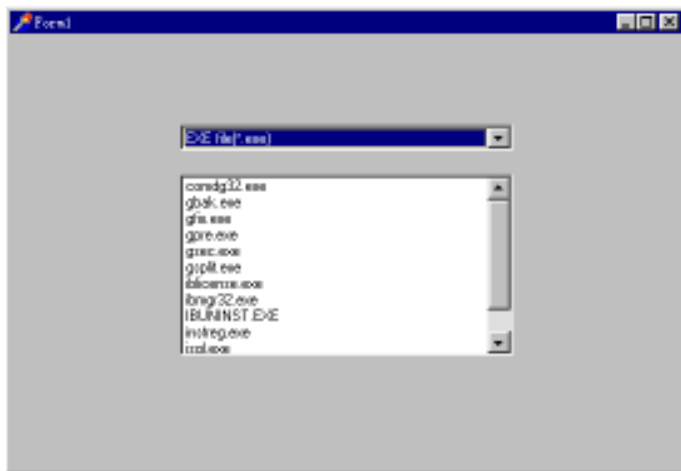


图 7-11 程序运行结果

在程序的运行过程中，用户可以选择显示文件的类型，如图 7-11 所示即为显示为以*.exe 结尾的可执行文件，当然用户也可以选择其它的文件筛选方式，如图 7-12 所示即为显示所有的以*.dll 结尾的动态链接库文件的程序运行结果。

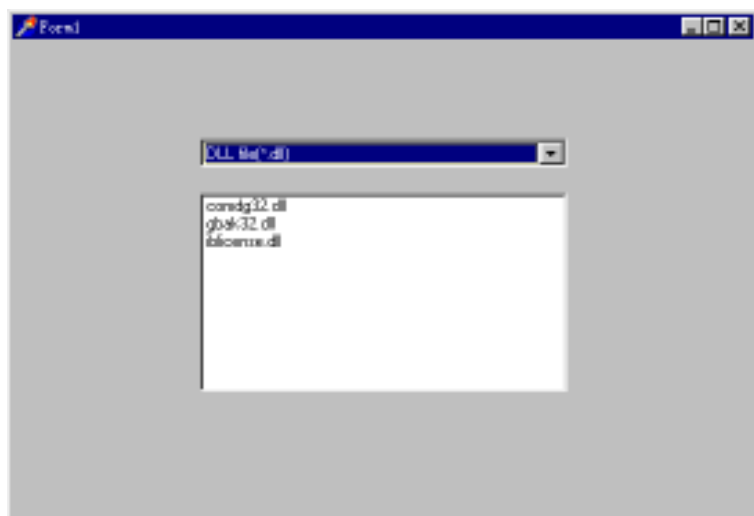


图 7-12 显示 DLL 文件

7.2 利用控件管理文件

前面分别的介绍了文件类控件中常用的四种 ActiveX 控件，但是到现在为止，各个控件之间还是独立运行，还没有使它们协调的工作，下面将通过一个综合应用的示例程序来说明如何使得四个文件类控件协调工作。

1. 开始工作

首先启动一个新的项目，选择菜单“File”中的“New Application”项，在 Delphi 的集成开发环境中就会弹出一个新建的窗体，在控件工具栏上选择 FileListBox 控件、Edit 控件、DriveComboBox 控件、DirectoryListBox 控件和 FilterComboBox 控件，并且把它们拖放到空白的窗体上，添加控件后的窗体如图 7-13 所示。